

Perguntas comuns e suas respostas:

- P: Tenho uma dúvida na questão tal.
R: A compreensão do enunciado faz parte da prova.
- P: O que será corrigido?
R: A lógica, a criatividade, a sintaxe, o uso correto dos comandos e dos tipos, os nomes das variáveis, a indentação, uso equilibrado de comentários no código e, evidentemente, a clareza. Nesta prova, você deverá sobretudo es-

crever códigos modulares, usando corretamente funções e/ou procedimentos, conforme o caso, além de uso correto de variáveis locais ou globais e a passagem de parâmetros por referência ou por valor.

- P: Posso fazer a lápis?
R: Não. A prova deverá ser feita a caneta.
- P: Posso responder na folha de questões?
R: Não. A prova deverá ser respondida na folha de respostas.

(10 pontos) 1) O que é algoritmo?

(09 pontos) 2) Diferencie Lista, Fila e Pilha.

(18 pontos) 3) Mostre a situação da fila F. Inicialmente vazia, após a execução de cada uma das operações a seguir:

```

1 Enfila(&F, 'a');
2 Enfila(&F, 'a');
3 Enfila(&F, 'b');
4 Enfila(&F, 'c');
5 Desenfila(&F);
6 Enfila(&F, 'e');
7 Enfila(&F, Espia(&F));
8 Enfila(&F, Desenfila(&F));
9 Desenfila(&F);
10 Enfila(&F, Espia(&F));
11 Enfila(&F, Enfila(&F, 'g') + 'a');
12 Enfila(&F, Desenfila(&F) + Espia(&F));

```

- A função 'Espia' retorna o elemento que seria desenfilado, mas não desenfila o elemento;
- A função 'Desenfila' retorna o elemento desenfilado;
- A função 'Enfila' retorna 1 quando consegue enfileirar e 0 quando não consegue.

– Você pode assumir que a fila, neste exercício, possui tamanho suficiente para enfileirar todos os elementos.

(18 pontos) 4) Mostre a situação da Pilha P. Inicialmente vazia, após a execução de cada uma das operações a seguir:

```

1 Empilha(&P, 'a');
2 Empilha(&P, 'b');
3 Empilha(&P, 'c');
4 Desempilha(&P);
5 Empilha(&P, 'e');
6 Empilha(&P, Topo(&P));
7 Empilha(&P, Desempilha(&P));
8 Desempilha(&P);
9 Empilha(&P, Empilha(&P, Empilha(&P, Topo(&P))) + 'd');
10 Desempilha(&P);
11 Empilha(&P, Desempilha(&P) + Desempilha(&P));

```

- A função ‘Topo’ retorna o elemento que seria desempilhado, mas não desempilha o elemento;
- A função ‘Desempilha’ retorna o elemento desempilhado;
- A função ‘Empilha’ retorna 1 quando consegue empilhar e 0 quando não consegue.
 - Você pode assumir que a pilha, neste exercício, possui tamanho suficiente para enfileirar todos os elementos.

(20 pontos) 5) Suponha que queremos decidir se uma dada sequência de parênteses, colchetes e chaves estão bem-formadas (ou seja, parênteses, colchetes e chaves são fechados na ordem inversa àquela em que foram abertos). Por exemplo, a primeira e a segunda das sequências abaixo estão bem-formadas enquanto a terceira e a quarta não estão.

Primeira cadeia - So when I die (the [first] I will see in (heaven) is a score list).

Segunda cadeia - ([((([[]) () (())]))]).

Terceira cadeia - Help(I[m being held prisoner in a fortune cookie factory]).

Quarta cadeia - Half Moon tonight (At least it is better than no Moon at all).

Suponha que uma frase qualquer que possua sequência de letras, pontos, espaços, parênteses, colchetes e chaves está armazenada em uma cadeia de caracteres (string) s. Como é hábito em C, o último caractere da cadeia é o caractere nulo.

Escreva a função `int bem_formada (char *string)` que retorna 1 se a string está bem formada, senão retorna 0. Implemente também, quaisquer funções auxiliares necessárias para o funcionamento da função `bem_formada()`.

(25 pontos) 6) Durante o desenvolvimento de um sistema uma fila foi utilizada para armazenar os números referentes ao processamento, porém quando o processamento é encerrado é necessário que seja impresso na tela todos os itens dessa fila ao contrário. Implemente uma função `void imprime-fila-ao-contrario()`, que imprime na tela o conteúdo da fila ao contrário da ordem armazenada. Você pode utilizar apenas as funções de Fila para manipular os elementos, ou seja, **NÃO** pode acessar os elementos da fila diretamente. Segue o protótipo das funções que estão disponíveis e que podem ser utilizadas:

- `void enfileira(fila_st *f, tipof novo)`
- `tipof desenfileira(fila_st *f)`
- `int esta-vazia(fila_st *f)`

Exemplo:

Imagine uma fila com os elementos armazenados:

10 20 -3 -15 20 50

Deve aparecer na tela:

50 20 -15 -3 20 10